Exploiting the full potential of JASigning to build an avatar signing train announcements

Ebling, S ; Glauert, J

Abstract: We are building a system that translates German train announcements of the Swiss Federal Railways (Schweizerische Bundesbahnen, SBB) into Swiss German Sign Language (Deutschschweizerische Gebärdensprache, DSGS) in real time and displays the result via an avatar. We use the JASigning system to animate the avatar. Deliverables of the projects during which JASigning was developed are the main source of documentation for the system along with notes on the website. Not all planned features have been fully implemented: some because they are used very infrequently; others because there is insuffcient linguistic research on which to base an implementation. Our team of hearing and Deaf researchers identified the avatar functionality we needed for our project. In close collaboration with the developers of JASigning, we then found workarounds for those features that were not yet directly available in the system. Our goal for this paper is to share the solutions we found, thereby providing a use case for exploiting the full potential of JASigning. The features we required were not specific to train announcements: Among them were the extension of non-manual features over multiple signs, the addition of pauses between items of a signed list, or the introduction of stamping movements following successive identical digits and fingerspelled letters. Hence, knowledge of how to achieve their designated effects in the JASigning system can be useful to persons working with other types of sign language data as well.

Originally published at:
Ebling, S; Glauert, J (2013). Exploiting the full potential of JASigning to build an avatar signing train announcements. In: Third International Symposium on Sign Language Translation and Avatar Technology, Chicago, IL, USA, 18 October 2013 - 19 October 2013.

Exploiting the Full Potential of JASigning to Build an Avatar Signing Train Announcements

Sarah Ebling
Institute of Computational Linguistics
University of Zurich
8050 Zurich, Switzerland
ebling@cl.uzh.ch

John Glauert
School of Computing Sciences
UEA Norwich
Norwich NR4 7TJ, UK
j.glauert@uea.ac.uk

ABSTRACT

We are building a system that translates German train announcements of the Swiss Federal Railways (*Schweizerische Bundesbahnen*, SBB) into Swiss German Sign Language (*Deutschschweizerische Gebärdensprache*, DSGS) in real time and displays the result via an avatar. We use the JASigning system to animate the avatar. Deliverables of the projects during which JASigning was developed are the main source of documentation for the system along with notes on the website. Not all planned features have been fully implemented: some because they are used very infrequently; others because there is insufficient linguistic research on which to base an implementation. Our team of hearing and Deaf researchers identified the avatar functionality we needed for our project. In close collaboration with the developers of JASigning, we then found workarounds for those features that were not yet directly available in the system. Our goal for this paper is to share the solutions we found, thereby providing a use case for exploiting the full potential of JASigning. The features we required were not specific to train announcements: Among them were the extension of non-manual features over multiple signs, the addition of pauses between items of a signed list, or the introduction of stamping movements following successive identical digits and fingerspelled letters. Hence, knowledge of how to achieve their designated effects in the JASigning system can be useful to persons working with other types of sign language data as well.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language generation*; K.4.2 [**Computers and Society**]: Social Issues—*Assistive technologies for persons with disabilities*

General Terms

Human Factors, Languages

Figure 1: JASigning character *Anna*

1. INTRODUCTION

Sign language avatars are virtual signers that provide access to information for Deaf[1] individuals. It is clear that sign language avatars cannot and should not replace human sign language interpreters, although this is a fear often expressed by Deaf signers. Instead, the aim should be for both forms of signing to co-exist and be used for different purposes: Interpreters are needed where sign language rendering has to be as accurate as can be (e.g., at a doctor's appointment) and where the human component plays an important role. Sign language avatars are suitable for providing an anonymized representation of a signer. Automatically animated sign language avatars, in addition, are able to render dynamic content, e.g., display the sign language output of a machine translation system or present the contents of a sign language wiki [4].

Several sign language avatars have been created in the past years. Among them is the Java Avatar Signing (JASigning) system [7, 6, 5, 9, 11, 12][2] developed during several interna-

[1] It is a widely recognized convention to use the upper-cased word 'Deaf' for describing members of the linguistic community of sign language users and, in contrast, to use the lower-cased word 'deaf' when describing the audiological state of a hearing loss [14].

[2] `http://vh.cmp.uea.ac.uk/index.php/JASigning`

tional projects (ViSiCAST,[3] eSIGN,[4] and DictaSign[5]). The main release of JASigning is freely available for research purposes and currently offers three different avatars, including the *Anna* character shown in Figure 1. Other characters have been developed for specific projects.

We are building a system that translates German train announcements into Swiss German Sign Language (*Deutschschweizerische Gebärdensprache*, DSGS) in real time and displays the result via an avatar. We use the JASigning system to animate the avatar. Deliverables of the projects mentioned above are the main source of documentation for the system along with notes on the website. Not all planned features have been fully implemented: some because they are used very infrequently; others because there is insufficient linguistic research on which to base an implementation.

Our team of hearing and Deaf researchers identified the avatar functionality we needed for our project. We also carried out a focus group study with seven Deaf signers to obtain further feedback. In close collaboration with the developers of JASigning, we then found workarounds for those features that were not yet directly available in the system. Our goal for this paper is to share the solutions we found, thereby providing a use case for exploiting the full potential of JASigning. The features we required were not specific to train announcements. Hence, knowledge of how to achieve their designated effects in the JASigning system can be useful to persons working with other types of sign language data as well, such as weather reports or customer information of different kinds.

The remainder of this paper is structured as follows: Section 2 gives an overview of the JASigning system. In Section 3, we present our project as part of which we are using the system. Section 4 describes the avatar functionality we needed for our project and presents our solutions for adapting JASigning to provide what we required. In Section 5, we give an overview of the issues we tackled and offer an outlook on further research questions.

2. JASIGNING

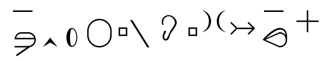
As its input, the JASigning system requires signs notated in the Hamburg Notation System for Sign Languages (HamNoSys) [16]. HamNoSys consists of approximately 200 symbols. It takes explicit account of the sublexical components hand shape, hand position (with extended finger direction and palm orientation as sub-components), location, and movement. An XML representation for HamNoSys exists as the Signing Gesture Markup Language (SiGML) [8]. Figure 2 shows the HamNoSys notation of the sign LAUTSPRECHER ('LOUDSPEAKER') in DSGS along with the corresponding SiGML code for the manual part of the sign. The sign is performed by opening and closing the dominant hand next to the ear.

Apart from information about the manual components of a sign, SiGML code may also contain information about non-manual features, i.e., about mouthings, mouth gestures,

[3]http://www.visicast.cmp.uea.ac.uk/

[4]http://www.visicast.cmp.uea.ac.uk/eSIGN/

[5]http://www.dictasign.eu/



```
<sign_manual>
  <handconfig ceeopening="slack" handshape="ceeall"
  mainbend="bent"/>
  <handconfig extfidir="u"/>
  <handconfig palmor="l"/>
  <location_bodyarm contact="close" location="head"
  second_location="ear" second_side="right_beside"
  side="right_beside"/>
  <rpt_motion repetition="fromstart">
      <tgt_motion>
      <changeposture/>
      <handconfig handshape="pinchall" mainbend="bent"/>
      </tgt_motion>
  </rpt_motion>
</sign_manual>
```

Figure 2: HamNoSys notation and corresponding SiGML code for the manual components of the sign LAUTSPRECHER ('LOUDSPEAKER') in DSGS

```
<hamgestural_sign gloss="LAUTSPRECHER">
  <sign_nonmanual>
    <mouthing_tier>
      <mouth_picture picture="laUtSprEC@r"/>
    </mouthing_tier>
  </sign_nonmanual>
  <sign_manual>
    <handconfig ceeopening="slack" handshape="ceeall"
    mainbend="bent"/>
    <handconfig extfidir="u"/>
    <handconfig palmor="l"/>
    <location_bodyarm contact="close" location="head"
    second_location="ear"
    second_side="right_beside" side="right_beside"/>
    <rpt_motion repetition="fromstart">
      <tgt_motion>
        <changeposture/>
        <handconfig handshape="pinchall" mainbend="bent"/>
      </tgt_motion>
    </rpt_motion>
  </sign_manual>
</hamgestural_sign>
```

Figure 3: SiGML code for the manual components and non-manual features of the sign LAUTSPRECHER ('LOUDSPEAKER') in DSGS

non-oral non-manual features (eyebrows, eye gaze, eyelids, nose), and non-facial non-manual features (head, spine, shoulders). For mouthings, transcriptions in the Speech Assessment Methods Phonetic Alphabet (SAMPA) [19], an ASCII version of the International Phonetic Alphabet (IPA), can be provided. As an example, the mouthing /Lautsprecher/ ('loudspeaker') is notated in SAMPA as `laUtSprEC@r`.[6]

Mouth gestures, non-oral non-manual features, and non-facial non-manual features are available in SiGML through alphanumeric tags, e.g., `L04` for pursed lips, `RB` for raised eyebrows, or `NO` for head nod [10]. Figure 3 shows the SiGML code for both the manual components and the non-manual features of the sign LAUTSPRECHER ('LOUDSPEAKER') in DSGS. Information about non-manual features is given inside a `<sign_nonmanual>` element. The SAMPA transcription of /Lautsprecher/ is specified inside a `<mouth_picture>` element. Along with this, the code for the manual components is provided inside a `<sign_manual>` element (cf. Figure 2).

Two SiGML variants exist: HNS SiGML and Gestural SiGML. Shown in Figures 2 and 3 is Gestural SiGML. This variant allows more fine-grained specification on the level of entire signs, the manual components, and the non-manual features of a sign. For example, each sign (represented as a `<hamgestural_sign>` element) may carry three attributes: `duration`, `speed`, and `timescale`. In addition, each non-manual tier element (such as `<mouthing_tier>` shown in Figure 3) may contain a child element `<..._par>` (e.g., `<mouthing_par>`) that causes the non-manual features embedded in it to be executed in parallel rather than in sequence.[7] Each non-manual tier element may also carry an attribute `presynchronization` or `postsynchronization` to control the synchronization of the non-manual features within it. Moreover, an attribute `fitpicturetomanual` can be specified for the `<mouthing_tier>` element to synchronize the duration of the mouthing and the manual activity of a sign. A mouthing can also be held or stretched over multiple signs with the `<mouth_meta>` element. Similarly, the `<hamgestural_segment>` element allows non-manual features to be applied to multiple signs. Figure 4 displays schematic HNS SiGML and Gestural SiGML code. Printed in bold are the elements and attributes described that are available only in Gestural SiGML.

The final Gestural SiGML code of a sign or sign sequence is handed over to the AnimGen animation engine [12], which generates motion data that can be used for the chosen avatar. Apart from SiGML code, AnimGen requires the input of four files defining the physical appearance of the avatar, since the motion data must be different for avatars with different dimensions. One of these files controls the non-manual features. It contains mappings of SiGML alphanumeric tags (like `L04`, `RB`, and `NO` above) to morph targets, which are points on the facial mesh that may be deformed. Each morph target carries the attributes `name` (e.g., `HPSF`), `amount`, and `timing`. The `amount` attribute specifies the amplitude of the morph, normally ranging between 0.0 and 1.0. The `timing` attribute consists of tags that control

- whether the morph is anchored to the start of the interval during which it is played;
- how long the attack time is;
- how the attack time is performed;
- how long the sustain time is;
- how long the release time is;
- how the release is performed; and
- whether the morph is anchored to the end of the interval during which it is played [11].

Hence, the attribute contains information about both the duration and the speed of a morph target.

New morph targets have been created with the ARP Toolkit[8] and added to the set of non-manual features available in JASigning (Robert Smith, personal communication). An `<avatar_morph>` element allows users to specify morph targets directly in SiGML code. An example is the following line in Figure 4: `<avatar_morph movement="HPSF" amount="2.0" timing="x m t m s l x"/>`.

The `sigml` root element may optionally contain a child element `<player_settings>` in which the avatar character and the camera perspective can be specified. With this, it is possible to switch avatars within a SiGML sequence.[9]

Some of the SiGML elements and attributes in the examples above are not yet fully functional in JASigning. In Section 4, we describe the avatar features we needed for our project and the workarounds we chose to achieve their designated effects. In what follows, we present our project as part of which we are using an avatar in more detail.

3. MACHINE TRANSLATION OF GERMAN TRAIN ANNOUNCEMENTS INTO SWISS GERMAN SIGN LANGUAGE

Deaf people today still face substantial barriers when using public means of transportation. Despite legal obligations in Switzerland to ensure accessibility for disabled people, much remains to be done in this area. For example, at railway stations, a considerable amount of information for passengers is conveyed via loudspeaker only and is not displayed in written form on the panels above the tracks. This makes it difficult for Deaf persons to know when a train is delayed or cancelled. A Deaf individual also has no access to announcements made by the loudspeakers on the trains.

[6] Note that syllable and accent information has been omitted from the SAMPA notation. It is ignored by the SiGML interpreter in JASigning, as it makes little difference to the visual appearance.

[7] Apart from the `<mouthing_tier>` element shown in Figure 3, the SiGML document type definition (DTD) permits the elements `<facialexpr_tier>`, `<shoulder_tier>`, `<body_tier>`, `<head_tier>`, `<eyegaze_tier>`, and `<extra_tier>`.

[8] `http://vh.cmp.uea.ac.uk/index.php/ARP`

[9] The minimum version of JASigning required for this is 095k.

```
<sigml>
  <hns_sign>
    <hamnosys_nonmanual>
      <hnm_shoulder tag=""/>
      <hnm_body tag=""/>
      <hnm_head tag=""/>
      <hnm_eyegaze tag=""/>
      <hnm_eyebrows tag=""/>
      <hnm_eyelids tag=""/>
      <hnm_nose tag=""/>
      <hnm_mouthpicture picture=""/>
      <hnm_mouthgesture tag=""/>
      <hnm_extramovement tag=""/>
    </hamnosys_nonmanual
    <hamnosys_manual>
      ...
    </hamnosys_manual>
  </hns_sign>
</sigml>
```

```
<sigml>
  <hamgestural_segment>
    <hamgestural_sign duration="" speed="" timescale="">
      <sign_nonmanual>
        <shoulder_tier presynchronization="slight_delay|start_slightly_ahead"
        postsynchronization="lasts_longer|ends_before">
          <shoulder_par> // available for all non-manual tier elements
            <shoulder_movement movement=""/>
          </shoulder_par>
        </shoulder_tier>
        <body_tier>
          <body_movement movement=""/>
        </body_tier>
        <head_tier>
          <head_movement movement=""/>
          <avatar_morph movement="HPSF" amount="2.0" timing="x m t m s l x"/>
        </head_tier>
        <eyegaze_tier>
          <eye_gaze direction=""/>
        </eyegaze_tier>
        <facialexpr_tier>
          <eye_brows movement=""/>
          <eye_lids movement=""/>
          <nose movement=""/>
        </facialexpr_tier>
        <mouthing_tier fitpicturetomanual="true|false">
          <mouth_picture picture=""/>
          <mouth_gesture movement=""/>
        </mouthing_tier>
        <extra_tier>
          <extra_movement movement=""/>
        </extra_tier>
      </sign_nonmanual>
      <sign_manual>
        ...
        <tgt_motion duration="" speed="" timescale="">
          <directedmotion direction="o" size="small"/>
          <handconstellation contact="touch"/>
        </tgt_motion>
        ...
      </sign_manual>
    </hamgestural_sign>
  </hamgestural_segment>
</sigml>
```

Figure 4: Comparison between schematic HNS SiGML (left) and Gestural SiGML (right)

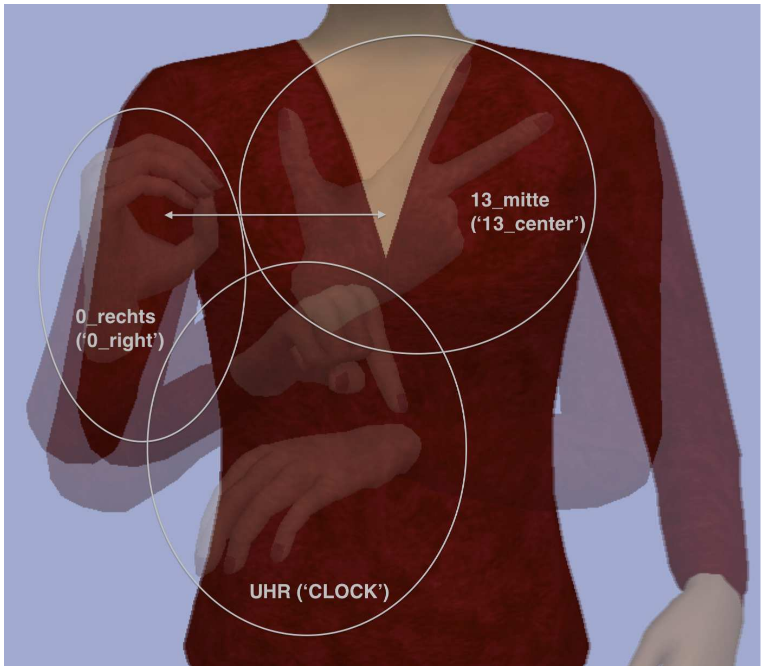


Figure 5: Locations of the sign sequence 13_mitte UHR 0_rechts 0_rechts ('13_center CLOCK 0_right 0_right') denoting the time specification *13:00*

A system has been built that converts French train announcements into French Sign Language (*Langue des Signes Française*, LSF) avatar animations and displays them on a monitor in a train station [17].[10] The system relies on parallel data consisting of written French announcements on the source side and LSF avatar animations on the target side, both as templates with slots, where slots can be, e.g., the names of train stations, types of trains, or reasons for delays. At runtime, the system identifies the template underlying the input segment and searches for the corresponding LSF avatar animation template. Subsequently, it fills the slots on the target side with the help of further written French–LSF avatar animation correspondences. However, simple concatenation is not enough: A coarticulation model is applied to ensure smooth transitions between surrounding and embedded animations. [17] performed a qualitative evaluation of the system and found that most users were satisfied with it. The users gave suggestions on how to further improve the system, e.g., through a more human-like appearance of the avatar. In addition, one participant proposed to make the avatar animations available not only on a monitor but also on a mobile phone.

We are building a system that automatically translates German train announcements of the Swiss Federal Railways (*Schweizerische Bundesbahnen*, SBB) into Swiss German Sign Language (*Deutschschweizerische Gebärdensprache*, DSGS). Our project team consists of two hearing and two Deaf researchers. DSGS is the sign language of the German-speaking area in Switzerland. It has approximately 6000 users [13] distributed across five dialects (Basel, Bern, Lucerne, St Gallen, Zurich). In our project, we focus on the Zurich dialect. To what extent DSGS is similar to German Sign Language is the subject of an ongoing study.

[10] Other customer service systems have been developed: For example, the TESSA system [3] and its successor VANESSA [18] translate a post office clerk's (spoken or written) utterances into British Sign Language.

The final output of our system is an avatar that signs the train announcements in real time on a mobile phone. The corresponding German text is shown as a subtitle beneath the avatar. Announcements remain available for a certain time so that they can be replayed. Hence, the target group of the application are not only Deaf and hard of hearing persons but also hearing persons looking for a replay functionality for train announcements.

Our approach differs from the work of [17] in that we do not work with templates nor pre-built avatar animations during the actual translation step. Given the standardized nature of train announcements, the approach of [17] is the most suitable for this type of data. However, our core research interest is in sign language machine translation, and our goal is to build a translation system that may later be extended to other domains with more lexical and syntactic variation. For the system at hand, we expect the output to be of good quality, due precisely to the standardized nature of our data. Note that this is not representative of the overall performance of sign language machine translation systems.

The input to the system are written announcements in electronic form, such as shown in Example 1. We deal with messages conveyed by loudspeakers at train stations, not in trains.[11]

(1) *Ausfallmeldung zum RegioExpress nach Olten. Der*

[11] The SBB use two different systems for this.

> *RegioExpress nach Olten, Abfahrt um 6 Uhr 41, fällt aus. Grund dafür ist eine technische Störung an der Lok.*
> 'Notice of cancellation of the regional express to Olten: The RegioExpress to Olten, scheduled to leave at 6:41, has been cancelled due to a technical problem with the locomotive.'

To obtain training, development, and test data for the machine translation system, we built a *parallel corpus* by manually translating a predefined number of German train announcements into DSGS. We received the German announcements from the SBB. To compile the parallel corpus, the hearing and Deaf members of our team

1. translated the written German train announcements into DSGS glosses;
2. signed the announcements in front of a camera on the basis of the gloss transcriptions;
3. notated the signs in the video recordings in HamNoSys (cf. Section 2);[12]
4. added information about non-manual features using SiGML alphanumeric tags; and
5. generated the avatar sequences from the resulting SiGML code to make sure that the quality of the manual translations was satisfactory.[13]

In what follows, steps 1 (gloss transcription) and 4 (non-manual feature annotation) are discussed in more detail.

The team developed several conventions for glossing to ensure consistency. For example, we defined the following sign string format for time specifications: <STUNDEN> UHR <MINUTEN> ('<HOUR NUMBER> CLOCK <MINUTE NUMBER>'). For train names, we used two different formats: If a commonly used abbreviation for a train name existed, we fingerspelled the letters of the abbreviation. This was the case, e.g., for *InterRegio* (IR) or *InterCity* (IC). In all other cases, we concatenated existing DSGS lexical signs; e.g., EURO ('EURO') and NACHT ('NIGHT') for *EuroNight*; STADT ('CITY'), NACHT ('NIGHT'), and LINIE ('LINE') for *CityNightLine*; or NACHT ('NIGHT') and VOGEL ('BIRD') for *Nightbird*.

We used lexical signs for widely known places such as Zurich, Basel, or Lucerne. For all other places (e.g., Sisikon, Wassen), we applied fingerspelling. Where several places co-occurred, we introduced a short pause after each: An example is the German announcement *Bus nach Wassen, Gurtnellen, Altdorf: Abfahrt auf dem Bahnhofplatz* ('Bus to Wassen, Gurtnellen, Altdorf: departure from the station square'), where we introduced a pause after the place names *Wassen*, *Gurtnellen*, and *Altdorf* in the corresponding DSGS translation.

[12]Where possible, they used the notations available in the DSGS database of [1].

[13]Note that for the machine translation step, the sign language side of the corpus is represented with glosses and HamNoSys notations.

We created two sets of HamNoSys notations for numbers between 0 and 60 with different location parameters: One set of numbers was signed in front of the body, the other slightly to the right. We then implemented a rule for time specifications according to which instances of <STUNDEN> ('<HOUR NUMBER>') were drawn from the first set (resulting in number signs performed in front of the signer's body), whereas instances of <MINUTEN> ('<MINUTE NUMBER>') were drawn from the second set (yielding a signing location slightly to the right). Hence, a time specification like *13:00* was glossed as 13_mitte UHR 00_rechts ('13_center CLOCK 0_right 0_right') and signed as shown in Figure 5. For train names like *S6* ('suburban railway no. 6'), the letter (*S*) was signed in front of the body and the number (*6*) slightly to the right. We also notated minute numbers between *00* and *09* as separate digits to have each digit signed individually (e.g., *00* as 'zero zero'), as is common for timetable information.

As a last step in the process of translating the German train announcements into DSGS, information was added about non-manual features, i.e., about mouthings and mouth gestures, head and shoulder movements, eyebrow movements, eyegaze, etc. Research on many European sign languages has shown that mouthings are not only capable of distinguishing between manual homonyms but also, when stretched over multiple signs, have an important prosodic function [2]. The mouthings used in DSGS are derived from Standard German rather than from one of the Swiss German dialects. We used the SAMPA notations of the Bonn Machine-Readable Pronunciation Dictionary for German (BOMP) [15]. The dictionary contains 141,230 entries. Where necessary, we modified them. Missing notations were also added.

4. GENERATING AN AVATAR SIGNING TRAIN ANNOUNCEMENTS WITH JASIGNING

We implemented the conventions described in the previous section using Gestural SiGML to make use of the full potential of the JASigning system. As stated in Section 2, some SiGML elements and attributes are not yet fully functional in JASigning. In close collaboration with the developers of JASigning, we created workarounds for them. For example, we needed the <hamgestural_segment> with which non-manual features can be applied to multiple signs. To replace its functionality, we modified the non-manual features we wanted to extend over an entire announcement in such a way that their morph targets were anchored to both the start and the end of a sign, using the **x** tag (cf. Section 2). We then applied the non-manual features to each sign of an announcement, as a result of which they seamlessly stretched over the entire announcement.

The `fitpicturetomanual` attribute (synchronizing the manual activity of a sign and the corresponding mouthing) could not be substituted easily. In our case, the duration of a mouthing mostly exceeded the duration of the corresponding manual activity. Therefore, we chose to speed up all mouthings slightly, as in the following example of /Lautsprecher/: <mouth_picture picture="laUtSprEC@r" speed="1.2"/>. At the same time, we decreased the speed of fingerspelled signs via the `speed` attribute of the <hamgestural_sign> element. Figure 6 shows an example of how this was done for the sign WASSEN. The example also shows how we

```
<hamgestural_sign gloss="WASSEN" speed="0.8">
  <sign_nonmanual>
    <mouthing_tier>
      <mouth_picture picture="vas@n" speed="1.2"/>
    </mouthing_tier>
  </sign_nonmanual>
  <sign_manual>
    <sign_manual> // W
      <handconfig bend5="hooked" handshape="finger2345"
      thumbpos="across"/>
      <handconfig extfidir="u"/>
      <handconfig palmor="d"/>
      <location_bodyarm location="shoulders"
      side="right_at"/>
    </sign_manual>
    <sign_manual> // A
      <handconfig handshape="fist"/>
      <handconfig extfidir="u"/>
      <handconfig palmor="d"/>
      <location_bodyarm location="chin"
      second_location="neck"
      second_side="right_beside"/>
    </sign_manual>
    <sign_manual> // S
      <handconfig handshape="fist" thumbpos="across"/>
      <handconfig extfidir="u"/>
      <handconfig palmor="d"/>
      <location_bodyarm location="shoulders"
      side="right_at"/>
      <directedmotion direction="o" size="small"/>
    </sign_manual>
    <sign_manual> // S
      <handconfig handshape="fist" thumbpos="across"/>
      <handconfig extfidir="u"/>
      <handconfig palmor="d"/>
      <location_bodyarm location="shoulders"
      side="right_at"/>
      <directedmotion direction="o" size="small"/>
    </sign_manual>
    <sign_manual> // E
      <handconfig handshape="flat" mainbend="hooked"
      thumbpos="across"/>
      <handconfig extfidir="u"/>
      <handconfig palmor="d"/>
      <location_bodyarm location="shoulders"
      side="right_at"/>
    </sign_manual>
    <sign_manual> // N
      <handconfig handshape="finger23" mainbend="bent"
      thumbbetween="45"
      thumbpos="across"/>
      <handconfig extfidir="uo" second_extfidir="o"/>
      <handconfig palmor="d"/>
      <location_bodyarm location="shoulders"
      side="right_at"/>
    </sign_manual>
  </sign_manual>
</hamgestural_sign>
```

Figure 6: SiGML code for the fingerspelled sign WASSEN in DSGS

dealt with successive identical fingerspelled letters: We introduced a short outward "stamping" movement with both occurrences of the letter (`<directedmotion direction="o" size="small"/>`). The same stamping movement was used for a succession of identical digits (e.g., *00* in the time specification *13:00*, cf. Section 3).

We added pauses between signs by inserting one or more `<nomotion/>` elements. We also caused the hands to return to a neutral position at the end of every signed announcement rather than to come to rest in the final posture of the announcement. We achieved this by adding an empty sign (i.e., `<hamgestural_sign>` element) at the end of the SiGML code of the announcement.

In addition, we adjusted and extended the inventory of non-manual features (cf. Section 2): For example, we modified the morph target mapping for the SiGML code `SH` (head shake) in such a way that it involved fewer movements of the head with higher amplitudes.

5. CONCLUSION AND OUTLOOK

In this paper, we have provided a use case for exploiting the full potential of the JASigning system. We used the system to animate an avatar that signs train announcements in Swiss German Sign Language (DSGS). We have described the avatar functionality we needed for our project: Among the features we required were the extension of non-manual features over multiple signs, the addition of pauses between items of a signed list, or the introduction of stamping movements following successive identical digits and fingerspelled letters. These features are not specific to train announcements. Hence, knowledge of how to achieve their designated effects in the JASigning system can be useful to persons working with other types of sign language data as well. Among the solutions we have presented are the manipulation of the anchoring behavior of morph targets and the introduction of artificial (empty) signs.

While we were able to find workarounds for most features that were not yet fully available in JASigning, one issue that remained was how to cause the non-manual features of a sign to precede the manual activity of the same sign. For example, our train announcements contained indexical (pointing) signs performed in a bottom corner of the signing space. The signs were accompanied by a movement of the head and the eyes towards the location of the indexical sign. In order for the signing to appear natural in this case, the onset of the non-manual features (head and eye movement) should precede the manual activity (indexical sign). Currently, we are unable to achieve this effect. That said, we believe the temporal coordination of the manual and non-manual components of a sign as a whole is an area that needs further linguistic research. New linguistic insights can then serve as the theoretical basis of an implementation in the JASigning system.

6. ACKNOWLEDGMENTS

The authors are grateful to Sandra Sidler-Miserez and Katja Tissi for their contribution to the project presented in this paper. They would also like to thank Penny Boyes Braem for giving valuable advice throughout this work and Thomas

Hanke for sharing his expertise in different areas of sign language technology.

7. REFERENCES

[1] P. Boyes Braem. A multimedia bilingual database for the lexicon of Swiss German Sign Language. *Sign Language & Linguistics*, 4:133–143, 2001.

[2] P. Boyes Braem. Functions of the mouthings in the signing of Deaf early and late learners of Swiss German Sign Language (DSGS). In P. Boyes Braem and R. Sutton-Spence, editors, *The hands are the head of the mouth: the mouth as articulator in sign languages*, volume 39 of *International Studies on Sign Language and Communication of the Deaf*, pages 99–133. Signum, Hamburg, 2001.

[3] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott. Tessa, a system to aid communication with deaf people. In *Proceedings of the Fifth International ACM Conference on Assistive Technologies*, Assets '02, pages 205–212, New York, NY, USA, 2002.

[4] E. Efthimiou, S.-E. Fotinea, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, and F. Lefebvre-Albaret. The Dicta-Sign Wiki: Enabling Web Communication for the Deaf. In K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, editors, *Computers Helping People with Special Needs*, volume 7383 of *Lecture Notes in Computer Science*, pages 205–212. Springer Berlin Heidelberg, 2012.

[5] R. Elliott, J. Bueno, R. Kennaway, and J. Glauert. Towards the Integration of Synthetic SL Animation with Avatars into Corpus Annotation Tools. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pages 84–87, 2010.

[6] R. Elliott, J. Glauert, J. Kennaway, I. Marshall, and E. Safar. Linguistic Modelling and Language-Processing Technologies for Avatar-based Sign Language Presentation. *UAIS (Universal Access in the Information Society) Journal*, 6:375–391, 2008.

[7] R. Elliott, J. Glauert, R. Kennaway, and K. Parsons. ViSiCAST Deliverable D5-2: SiGML Definition. Technical report, ViSiCAST project, 2001.

[8] R. Elliott, J. R. W. Glauert, J. R. Kennaway, and I. Marshall. The development of language processing support for the ViSiCAST project. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, Assets '00, pages 101–108, Arlington, Virginia, United States, 2000. ACM.

[9] J. Glauert and R. Elliott. Extending the SiGML notation – a Progress Report. In *Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, Dundee, Scotland, 2011.

[10] T. Hanke. ViSiCAST Deliverable D5-1: Interface Definitions. Technical report, ViSiCAST project, 2001.

[11] V. Jennings, R. Elliott, R. Kennaway, and J. Glauert. Requirements For A Signing Avatar. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.

[12] J. R. Kennaway, J. R. W. Glauert, and I. Zwitserlood. Providing signed content on the Internet by synthesized animation. *ACM Trans. Comput.-Hum. Interact.*, 14(3), Sept. 2007.

[13] M. P. Lewis, editor. *Ethnologue: Languages of the World*. SIL International, Dallas, Tex., 16th edition, 2009. Online version: `http://www.ethnologue.com/`.

[14] G. Morgan and B. Woll, editors. *Directions in sign language acquisition – Trends in language acquisition research*. John Benjamins Publishing Company, Amsterdam, 2002.

[15] T. Portele, J. Krämer, and D. Stock. Symbolverarbeitung im Sprachsynthesesystem Hadifix. In *Proceedings of the 6. Konferenz Elektronische Sprachsignalverarbeitung*, pages 97–104, Wolfenbüttel, 1995.

[16] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, and J. Henning. *HamNoSys: Version 2.0: Hamburg Notation System for Sign Languages. An introductory guide*. Signum, Hamburg, 1989.

[17] J. Segouat. *Modélisation de la coarticulation en Langue des Signes Française pour la diffusion*. PhD thesis, Université Paris Sud, 2010.

[18] J. Tryggvason. Vanessa: A system for council information centre assistants to communicate using sign language. `http://www.visicast.cmp.uea.ac.uk/eSIGN/Images/VANESSA.pdf`, 2004.

[19] J. Wells. SAMPA computer readable phonetic alphabet. In R. W. D. Gibbon, R. Moore, editor, *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, Berlin, Germany, 1997.